OC 3570
Juan Conforto

Geostrophic Velocity field obtained from CTDs vs. Velocity field measured from ADCP

INTRODUCTION

This work is based on data obtained during the OC3570 summer cruise on board R/V "Point Sur". The ship is owned by the National Science Foundation (NSF), and operated for the Central California Oceanographic Cooperative (CENCAL) by Moss Landing Marine Laboratories. The ship departed from Moss Landing at 1548Z on July 21. During the navigation CTD data was acquired at 31 different stations. The stations are grouped along three lines corresponding to lines 67, 70 and 77 of the California Cooperative Oceanic Fisheries Investigation (CalCOFI), forming what will be called the CenCal Box. Stations 1 to 10 correspond to CalCOFI line 67, stations 10 to 21 to Cal COFI line 70 and stations 21 to 31 to Cal COFI line 77.

It was projected to take the casts up to 1000 m (~1010 decibars), but for stations near the coast, they where taken up to the deepest the bottom allowed. That was the case for station 1, in front of Moss Landing, and stations 26 to 31, close to

Port San Luis. The first CTD cast, at station 1, started July 21 at 1630Z and at 0905Z July 24 the last cast was terminated at station 31.

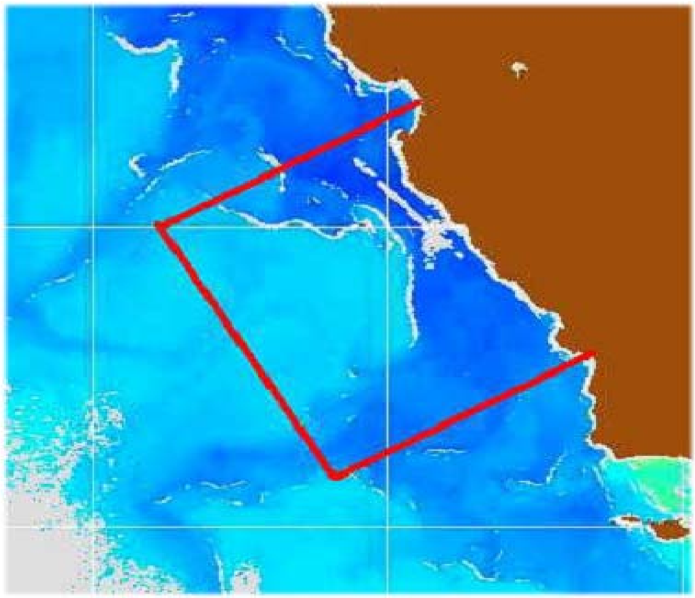
The instrument employed was the Sea-Bird's 911*plus* CTD, produced by Sea-Bird Electronics, Inc. This CTD is provided with one conductivity and one temperature sensor (fitted with a *TC Duct* and constant-flow pump), and an internally mounted, temperature-compensated Paroscientific Digiquartz pressure sensor for 6800 meter (10,000 psia) full scale range. This allowed obtaining salinity and temperature along with pressure profiles.

The other instrument employed during the cruise was the ADCP (Acoustic Doppler Profiler) from RD Instruments installed on board. This instrument allowed to compute the along-track and across-track velocities up to 450 m depth.

THE CALIFORNIA CURRENT

The oceanography of the cruise area is tied to processes of the California Current. The California Current is the eastward portion of the clockwise North Pacific Gyre, being therefore considered as an eastern boundary current. California Current transports low salinity, cool water toward the south. This current can be described as a broad, shallow, slow southward moving current and exhibits high spatial and temporal variability. The core of the California Current lies in the salinity minimum about 300 km offshore of Point Sur, and is not associated generally with a thermal gradient (Lynn et al. 1982). This makes location of the California Current difficult from infra-red imagery. The low salinity waters derive generally from the low salinities in the Gulf of Alaska.

Although this California Current was not wholly detected on the cruise area, there is also a northward flowing undercurrent associated with the coastal flow: the California Undercurrent. This California Undercurrent was detected on the cruise data flowing poleward in front of the coast between Port San Luis and Monterey Bay at depths between 150 and 500 m.



As was observed on satellite imagery and the CTD-derived geostrophic velocity field, the California Current is richly populated with semi-stationary jets and eddies. Satellite imagery shows cold filaments on the order of 50 km wide to extend offshore (Strub et al. 1991, Ramp et al. 1991a). The importance of these features, which represent the highly variable oceanographic "weather" of the California Current, lies in their offshore transport of cool, nutrient-rich upwelled water. This extends the effects of nearshore upwelling, which is confined to a band about 50 km wide, to several hundred km. In what are called "squirts," the flow may be directed offshore, and where the "squirt" dissipates elongated "hammerhead" features evolve. Between mesoscale eddies, the flow is directed offshore north of cyclonic eddies and onshore south of them.

GEOSTROPHIC VELOCITY

The Geostrophic method for computing the Velocity, as explained by Pickard and Emery (1990), is based on measurements of temperature and salinity profiles.

The Geostrophic Currents are the result of a balance of forces. Let's see this. The distribution of mass in the ocean can be obtained from the equation of state using salinity, temperature and pressure (depth). Actually, with this method we can observe the distribution of density rather than the distribution of mass. In general, density increases with depth, as pressure increases in the same way. The stable situation would correspond to a vertical distribution of increasing density, that is, no gradient of density in the horizontal direction. But if we find horizontal gradients of density, that will imply horizontal gradients of pressure. There is a parameter, called Dynamic Height, which will relate these pressure differences. This dynamic height, which will be normally reflected as different heights at the sea surface, represents the ability of a column of water to do work due to differences in geopotential. Changes in Dynamic Topography will then provide a value of the horizontal pressure gradient force.

Having a pressure gradient force we will have a motion. But since we have a motion we will have to take account of the Coriolis effect due to the rotation of the earth. Geostrophic currents will then be the result of the balance between the pressure gradient force and the Coriolis force. This balance can be expressed mathematically as:

$$V \cdot 2\Omega \sin\varphi = -\frac{1}{\rho}\frac{\partial\rho}{\partial x}$$

where V=speed of flow, Ω=angular speed of rotation of the earth, ϕ=latitude, ρ=water density and $\frac{\partial\rho}{\partial x}$=horizontal pressure gradient.

As explained before we do not measure density directly but rather we compute it from measures of salinity and temperature profiles. As we are looking for the pressure gradient in order to compute the geostrophic velocity field, we need to describe the pressure field somehow. The way we use is to give the geopotential height of equal pressure surfaces. This way we are obtaining the dynamic topography (the geopotential height) of equal pressure (equal to equal depth) surfaces. This dynamic topography field will describe clearly the pressure gradients. The geostrophic flow can be described relative to the geopotential height field as follows: in the northern hemisphere the higher sea surface (lighter water) is on the right of the direction of the geostrophic flow.

The geostrophic velocity at one level V_1 relative to a lower level V_2 between two stations denoted by A and B is estimated by

$$V_1 - V_2 = \frac{\left(\Delta\Phi_B - \Delta\Phi_A\right)}{2d\Omega Sin\phi_{ave}}$$

where d is the distance between stations and the geopotential anomaly ($\Delta\Phi$) is the integral of the specific anomaly (δ),

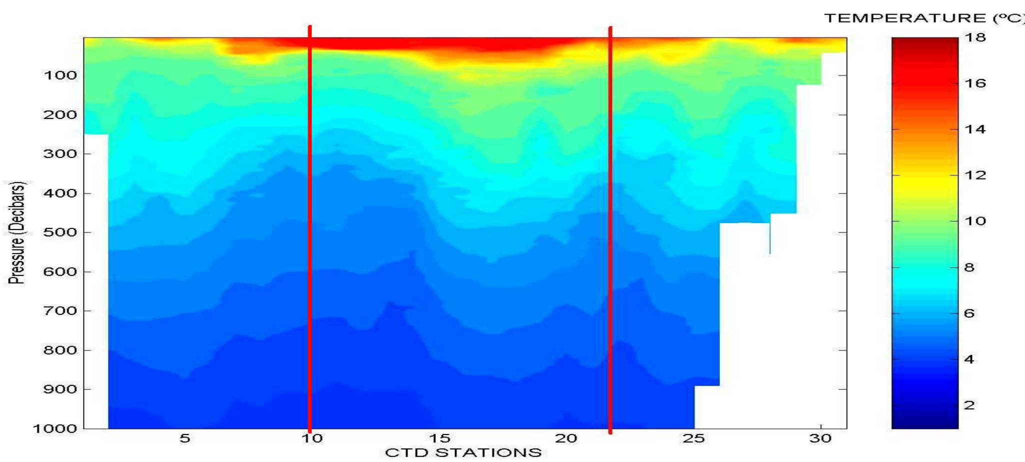
$$\Delta\Phi = \int_{P_1}^{P_2} \delta dp$$

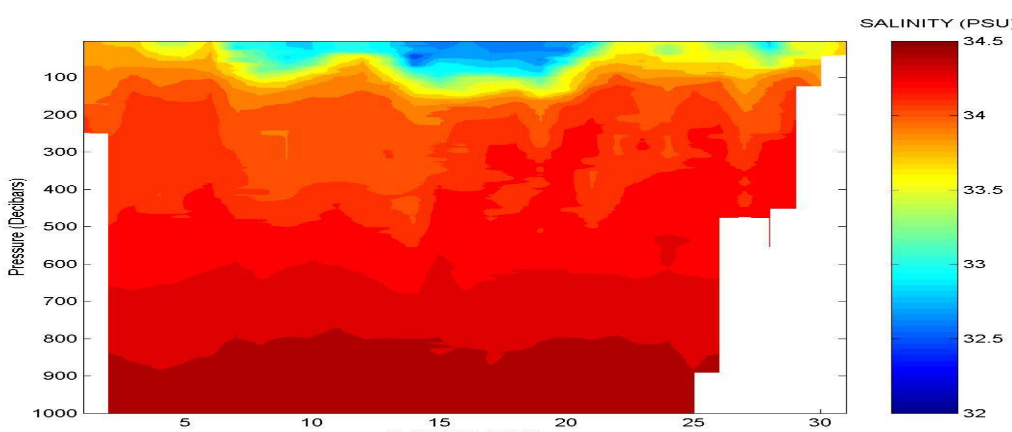
If we traverse the section from station A to station B, a positive velocity is directed to the left of the path, whereas a negative velocity is directed to the right. The big limitation of the geostrophic relation is that it only gives us relative currents. This means that we obtain currents at one level relative to another level. If we want absolute currents, we need to set the absolute current at some level that will be used as reference level. The most common method is to assume that the absolute current at some depth is zero and use that depth as reference level. That depth will be called the depth of no motion. After setting this depth of no motion, the geostrophic method will give us the absolute current at all other levels. The level of no motion, as explained, is chosen arbitrarily. For this work we initially used 1000 m as level of no motion.

On this work the SeaWater subroutines are used to compute geopotential anomalies, geostrophic velocities and distances between stations using data of the CTD casts.

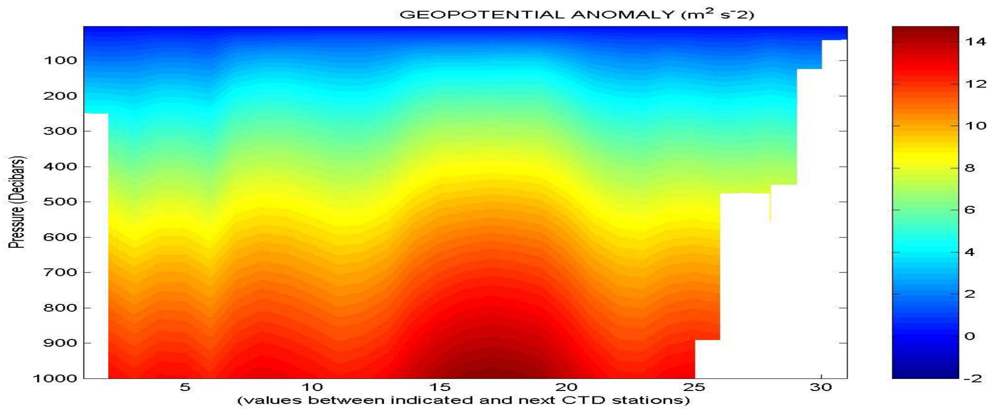
OBTENTION OF THE GEOSTROPHIC VELOCITY FIELD ALONG TRACK

As explained before, the geostrophic velocity was obtained comparing consecutive CTD station data. Using Matlab and the Seawater subroutines, salinity, temperature and pressure data are extracted from the CTD ASCII data files corresponding to each station. This information is grouped in matrices having a column for each station. A matrix for Temperature and another one for Salinity are produced this way. The location of each station – latitude and longitude – was also extracted. The distances between stations were computed and grouped in a vector. Pressure values corresponding to temperature and salinity data were also extracted (from CTD files) and grouped in another vector. As a result, temperature and salinity fields along track can be plotted.

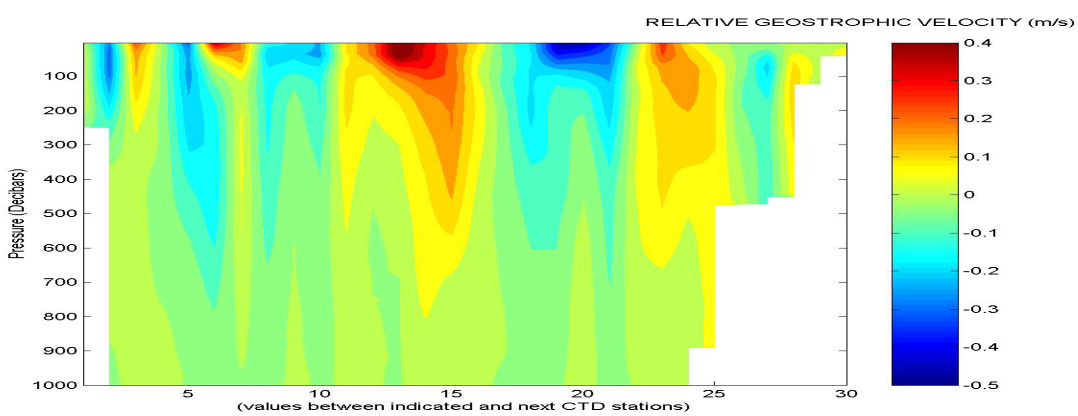




Once the salinity and temperature matrices were created, the geopotential anomaly between stations was obtained using the sw_gpan.m Seawater Matlab subroutine and was grouped in a matrix.



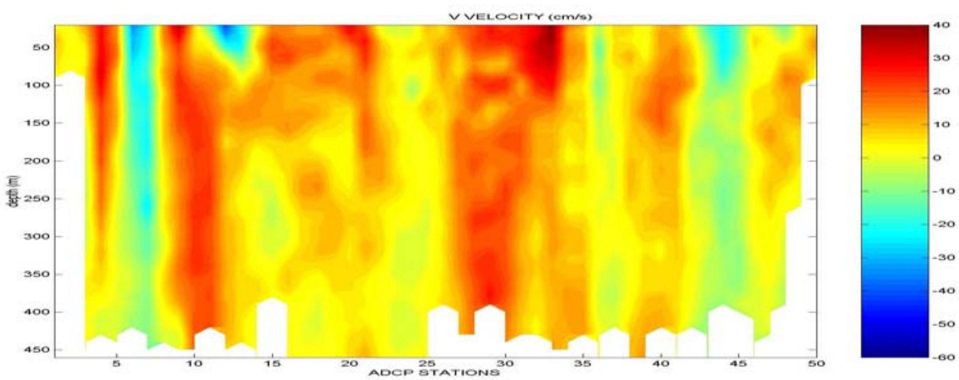
From the geopotential anomaly, the geostrophic velocity was obtained using the sw_gvel.m Seawater Matlab subroutine. The geostrophic velocity was initially computed using the surface as the no-motion reference. Then, this reference level is changed and a new velocity field was computed using 1000 db as level of no-motion.

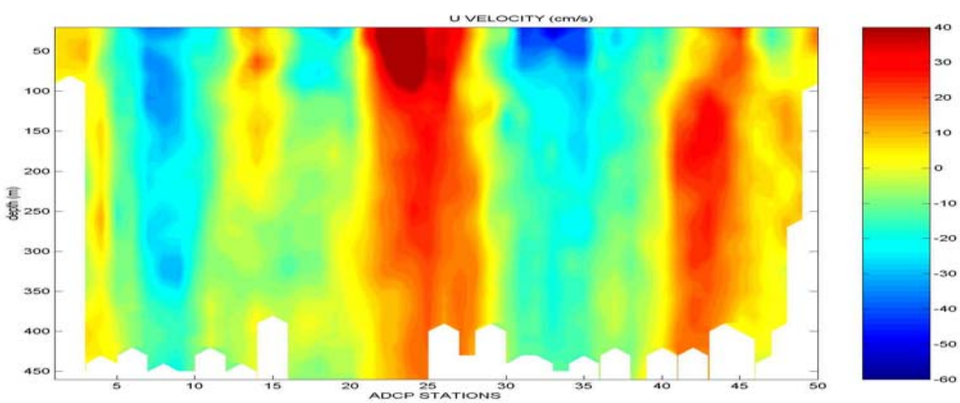


After computing the geostrophic velocity between stations, the velocities normal to the track can be represented together in a plot from station 1 to 31 and from surface to 1000 decibars. As explained above, this velocity field was computed assuming 1000 db is the level of no motion.

OBTENTION OF THE ADCP VELOCITY FIELD ALONG TRACK

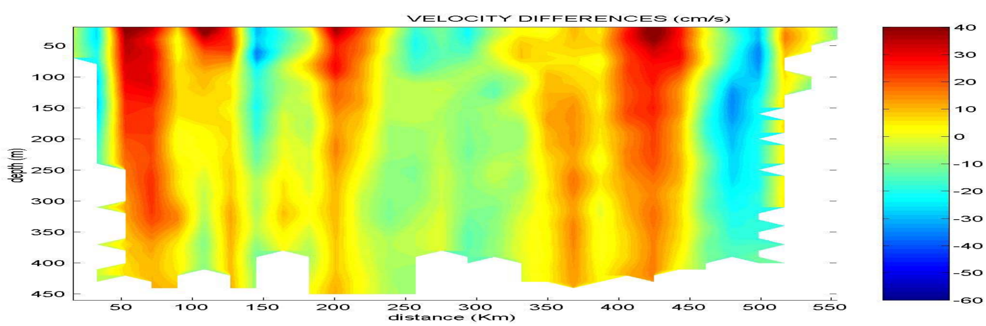
The ADCP instrument recorded the along-track and across-track water velocity. From the obtained files, using a Matlab program, both velocity data were extracted and grouped in a matrix, representing the velocity field along the track. This data allows representing the water velocity every 10 m depth up to 450 m. We have the across-track velocity represented by U and the along-track velocity represented by V.



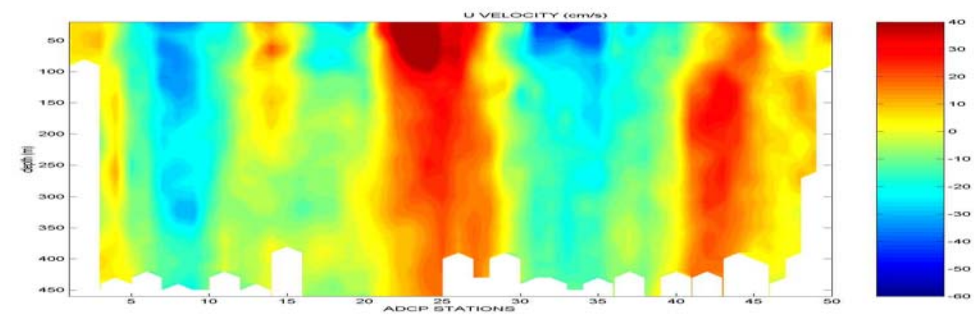


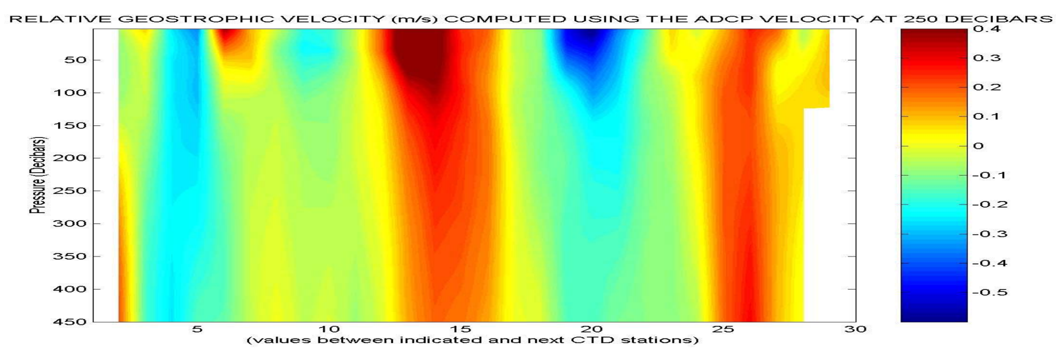
COMPARISON OF CTD GEOSTROPHIC VELOCITY AND ADCP VELOCITY

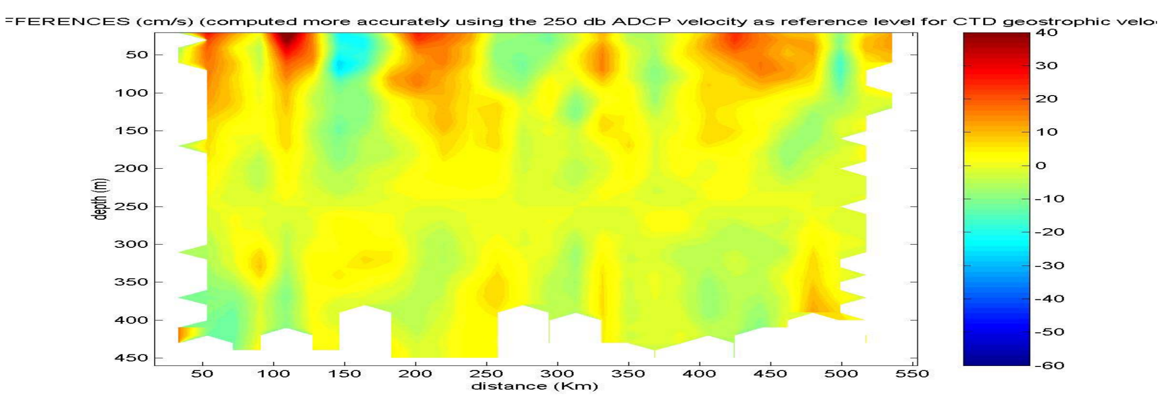
The main objective of this work was to compare the geostrophic velocity obtained from the CTD data with the velocity obtained from the ADCP instrument. The CTD velocity field was obtained up to 1000 db and the CTD velocity field up to 450 m. We have also to consider that CTD field correspond to 31 CTD stations (30 columns of velocities in the final matrix) and the ADCP have data for 50 different locations. The first step was then to refit both sets of data to compare them. Both them were refitted producing new matrices for CTD and ADCP data with the same rows and columns, by interpolating the original data matrices. As a result, the new matrices could be subtracted. After subtracting them we obtained a matrix of the differences, the comparison between both data sets.



The resulting plot showed large differences between both data sets. The main reason for these large differences was the reference level used to compute the geostrophic velocity field. As explained, the geostrophic velocity field was computed assuming 1000 db as level of no motion. That resulted to be a bad assumption. Assuming that the ADCP velocity data was accurate enough, the velocity obtained from it could be used as reference to compute a new more-accurate geostrophic velocity field. Observing the ADCP velocity data, the velocity corresponding to 250 m depth was selected as the reference velocity. Looking for the deepest velocity available for the whole data set, 250 m was selected because it was the deepest that covered almost the whole data set. As a consequence, a vector containing velocity data at 250 m depth was extracted from the ADCP velocity field and saved. This velocity vector was used now to compute a more accurate geostrophic velocity field. In the new method, 250 m was used initially as the level of no motion and the geostrophic velocity was computed. Then the ADCP velocity for 250 m depth was added to the corresponding velocity column, resulting a new geostrophic velocity matrix, which had for 250 m depth the same velocity values obtained from ADCP. This was the new, more accurate, geostrophic velocity field.







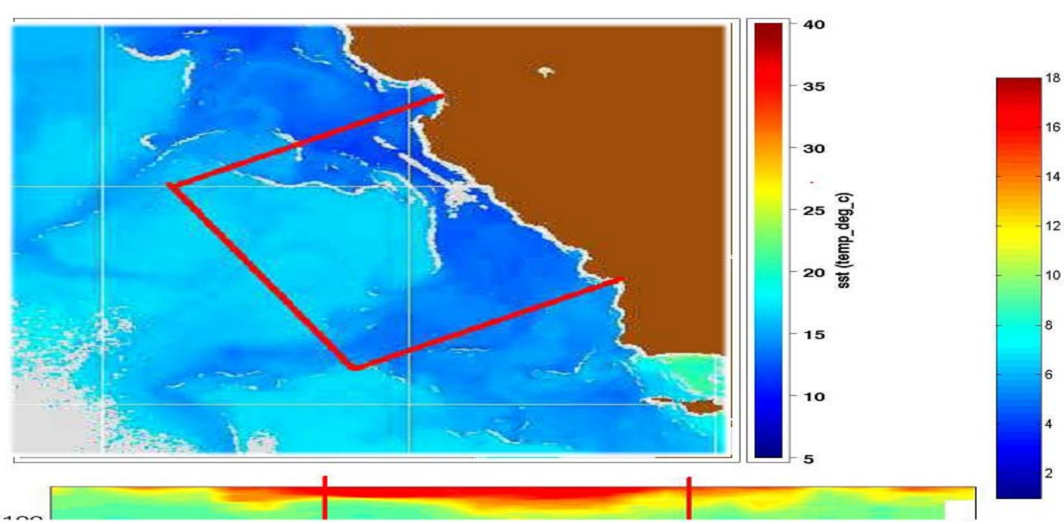
This new geostrophic velocity field looked more similar to the ADCP velocity field. Now, after subtracting both data sets, the new plot of differences looked quite different.

Now the plot has values of zero or very close to zero for most of the covered area. We can observe, however, some important differences for the upper part of the plot. As expected, the surface layer (up to about 150 m depth) moves not only because of the geostrophic balance; there are other forces that influence this surface layer motion. Among them we can cite Ekman transport, due to the wind, and also the motion due to tides.

COMPARISON OF SURFACE TEMPERATURE AND NOAA-17 SST

We can compare the temperature field obtained from CTD data and satellite Sea Surface Temperature (SST). For the duration of the cruise – 20-24 July – the low-level stratus clouds impeded getting SST values from satellite. The closest – in time – SST image available corresponds to July 17. The image was obtained after data from NOAA 17 satellite. Considering the slow-changing temperature features in the ocean compared to the 5 days offset, we can still get some conclusions comparing this image to the temperature field obtained from CTD data.

After drawing the approximate location of lines 67, 70 and 77 on the SST image, we can compare it with the temperature field.



For line 67 we observe colder SST temperatures shoreward. Reading the SST scale we can obtain temperatures moving from about 13 ºC, close to Moss Landing, to about 16.5 or 17ºC at the end of the line. The increase in temperature is not uniform along the line but rather oscillating, although an increasing trend is observed. Those values and oscillations agree with the data we observe at the surface of the CTD Temperature field.

For line 70 we can observe on the SST satellite image little variation of temperature. Temperature is around 17 ºC at the north end and around 15 ºC at the south end, with a region of cooler temperatures (~ 14 º C) right before reaching the south end. This observations and values fully agree with the CTD surface temperature data.

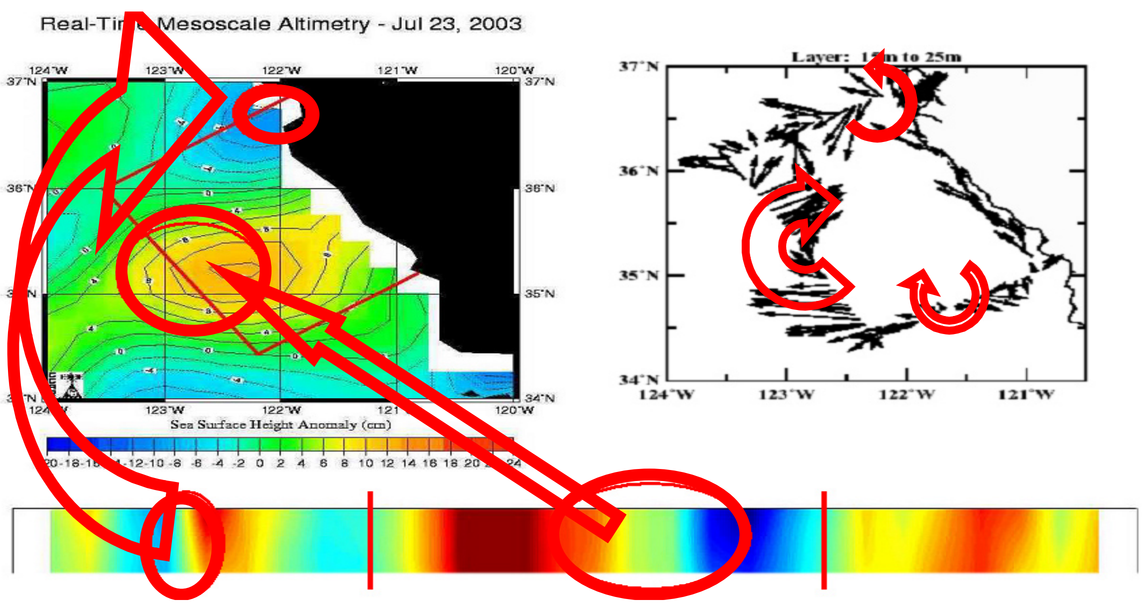
For line 70 we observe on the SST satellite image temperatures around 15 º C offshore and temperatures around 13 ºC on-shore, crossing a region of cooler temperatures (~ 12 º C) in the middle region of the line. All this fully agrees again with CTD surface temperature data.

COMPARISON OF GEOSTROPHIC VELOCITY FIELD AND TOPEX MESOSCALE ALTIMETRY

As explained before, in the northern hemisphere the lighter water – the higher water surface – is to the right of the geostrophic current. This allows us to compare the geostrophic velocity field and the satellite altimetry. Positive and negative geostrophic cross-line velocities must agree with the elevation slopes of the water surface obtained from TOPEX satellite altimetry. We must be aware, though, that the geostrophic velocity field along the navigation line tells us only the relative direction of the velocity. However, we don't know the actual angle between the geostrophic velocity vector and the navigation line.

Looking at the geostrophic velocity field for the line 67, we can observe that, starting from Moss landing, geostrophic velocity is weak but moves the water slightly southward (is slightly positive for our case), turning northward and changing strongly to the south and finally gets very weak or turns slowly to the north. Comparing this to the altimetry field, we observe that the navigation line crosses a low water surface region. This low level area will produce a counterclockwise geostrophic flow. This counterclockwise gyre will produce along the line 67 the effects described for the geostrophic flow. Therefore we observe that the geostrophic velocity field and the altimetry field agree for line 67.

For line 70, starting from the north, we have initially very weak geostrophic motion or slightly to the west. Then it changes becoming strongly eastward and



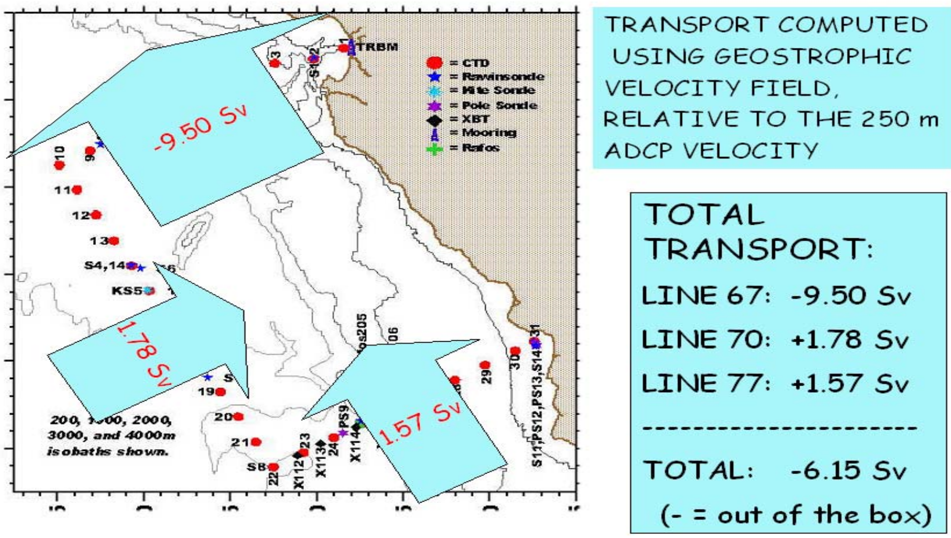
finally turning strongly westward. Looking at the altimetry field we observe that line 70 crosses the westward side of a seawater high elevation. This high water will produce a geostrophic clockwise flow. The west bound of this gyre, being crossed by the navigation line, will produce the same effects in strength and direction explained before for the geostrophic velocity field. Therefore we see that the altimetry field and the geostrophic velocity field agree for line 70.

For line 77, starting from the west, we have initially strong southward flow. Then it changes slowly becoming strongly northward. If we compare this with the altimetry field, we observe that the line crosses the south bound of a high elevation on the sea level. This will produce a clockwise gyre. The effects of this clockwise gyre would produce opposite geostrophic flow than described for the geostrophic velocity field. Therefore we see that there's no agreement for this line 67 between altimetry field and geostrophic velocity field.

GEOSTROPHIC VOLUME TRANSPORT

Having calculated the geostrophic velocity, the volume transport can be easily obtained. From the geostrophic velocity matrix we have the average velocity corresponding to the area formed by the distance between stations and elementary increase in depth. If we want to obtain the transport we just have to multiply the geostrophic transport by the corresponding area. This way we obtain a new matrix of transports. Putting all the length units in meters and the geostrophic velocity in meters per second, the transport will be in m^3/sec. If we divide by 10^6 we have the transport in Sverdrups (Sv).

After calculating the transport matrix by the explained method, we can extract the total transport across lines 67, 70 and 77. To obtain this we compute the corresponding sum of elements of the matrix corresponding to the columns of each line. This way we have a value of the transport across each of the lines 67, 70 and 77. The result of applying all this to the geostrophic velocity field obtained using the 250m-depth ADCP reference velocity is as indicated on the figure.

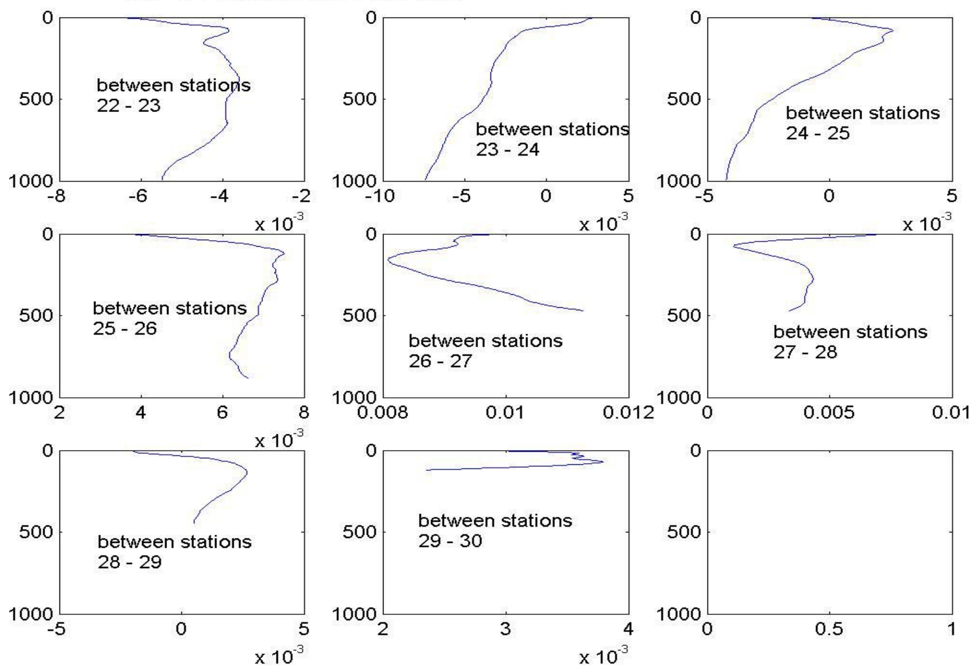


Because of the mass conservation principle, the volume of water entering the box must be equal to the volume of water leaving the box. As we can observe, the total transport is –6.15 Sv and not zero. The total transport would be zero if we had considered the whole depth for all the CTD stations. As the casts were up to 1000 m depth and did not reach the bottom depth for most of the stations, there is a large amount of volume transport not considered on this computation. Therefore the same total transport leaving the box between 1000 m depth and the surface must be somehow entering the box between 1000 m depth and the bottom.

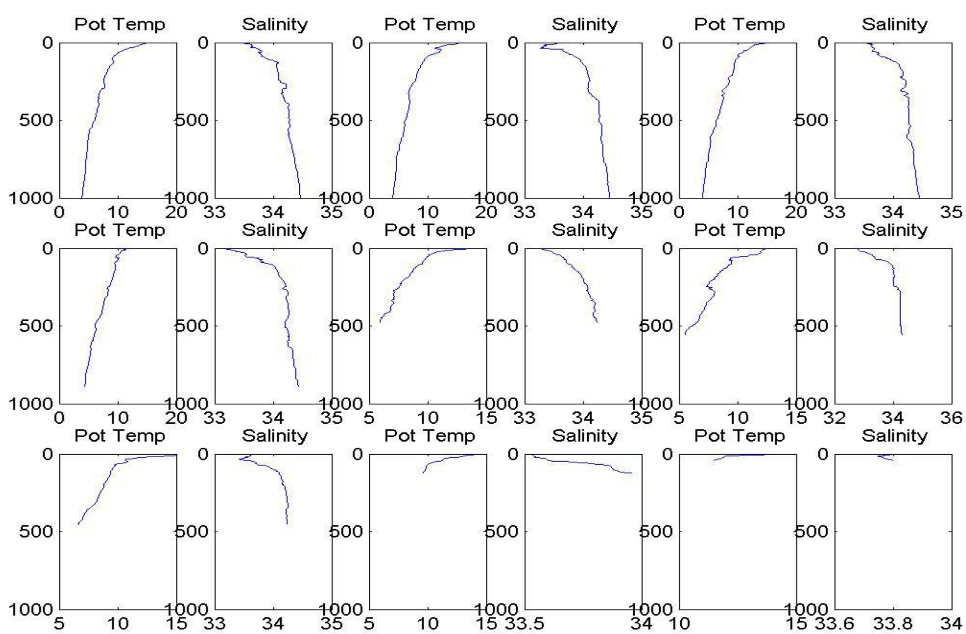
As we can observe on the figure, most of the transport is leaving the box across line 67, flowing poleward. We can also observe that the transport across line 77 is also poleward. Then, it seems that there must be a poleward flow along the coast and close to it.

Looking at the transport profiles for line 77 we can observe that close to the coast, there is strong poleward flow with the core at depths between 100 and 500 m. This flow must correspond to the California Undercurrent. We can find more evidence of this if we look at the Salinity and temperature profiles for line 77.

LINE 77: TRANSPORT PROFILES

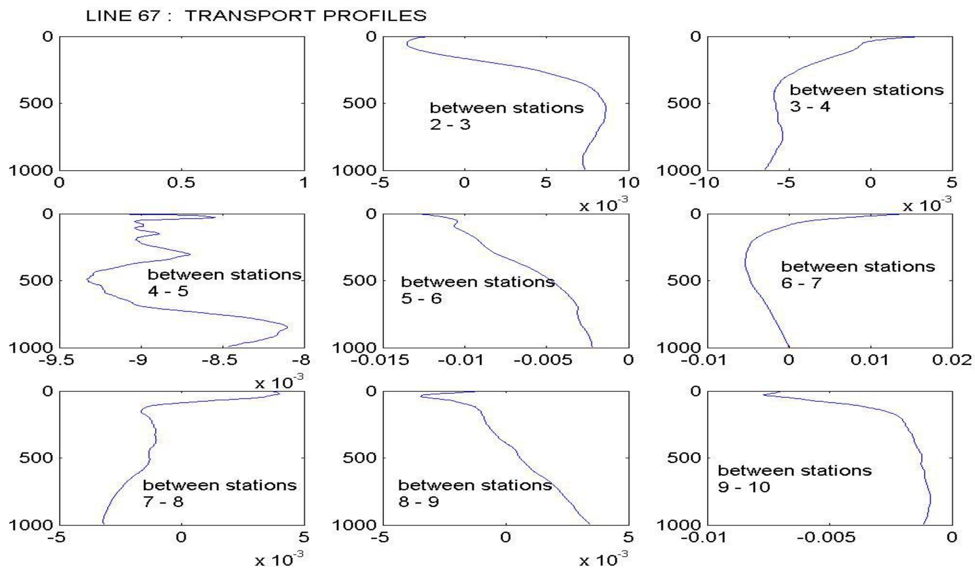


As we can see on the Potential Temperature and Salinity profiles for line 77, the water between 100 and 500 m is saltier and warmer over the continental slope than close to the coast. This is characteristic of the California Undercurrent: warmer and saltier water flowing to the north over the continental slope (Hickey, 1998).

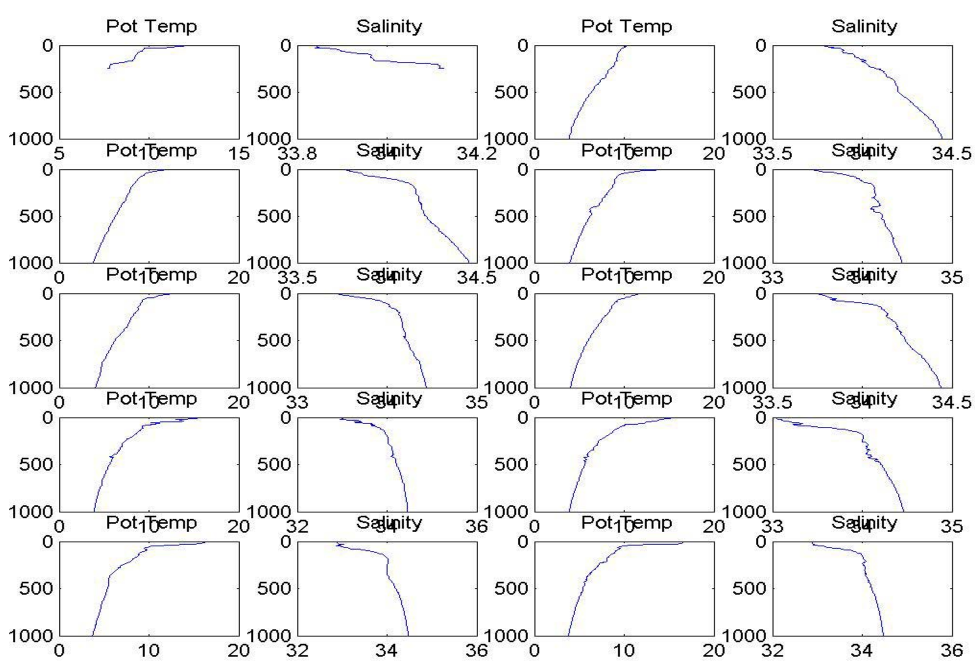


POTENTIAL TEMPERATURE AND SALINITY PROFILES FOR LINE 77

Looking at the transport profiles across line 67 we can observe a poleward flow below 200 m depth.



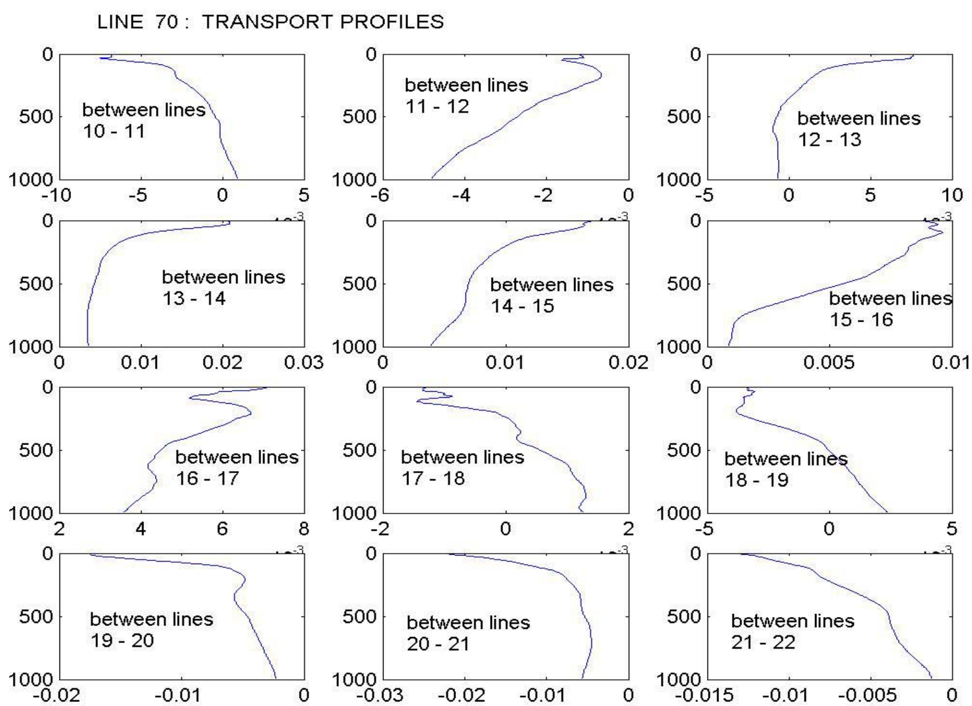
If we take a look at the Potential temperature and Salinity profiles across line 67 we can observe saltier and warmer water below 150 m in a similar way than line 77. This shows the same California Undercurrent characteristics.



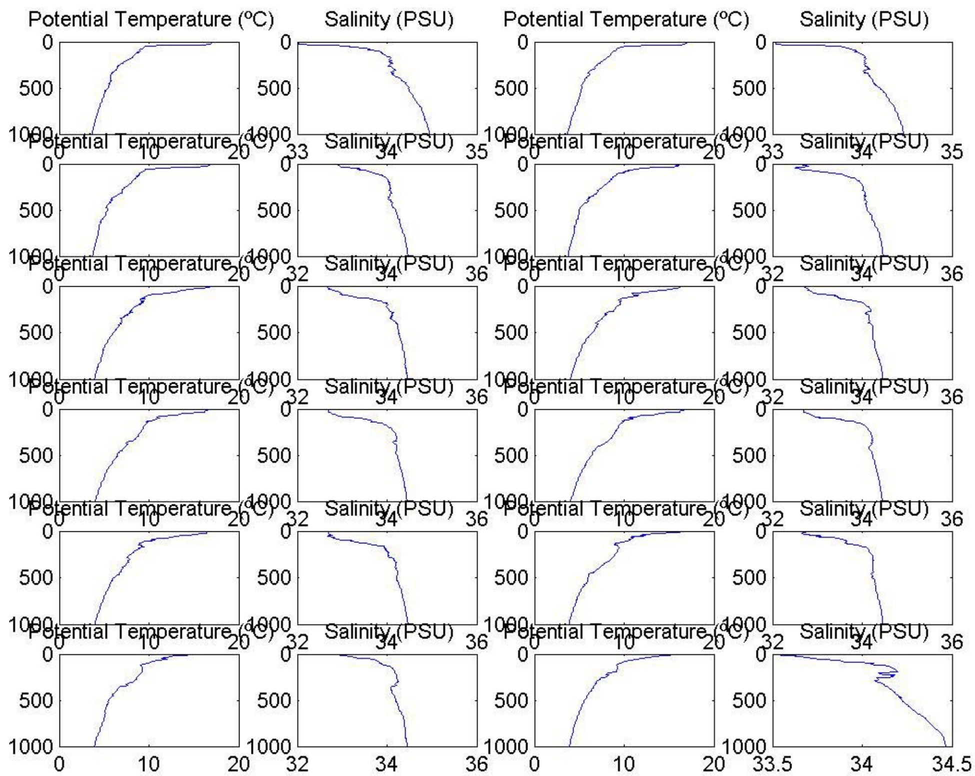
POTENTIAL TEMPERATURE AND SALINITY FOR LINE 67

Looking at line 70 we observe that, in general, this shoreward transport (below 150 m) increases with depth. This increase in transport with depth may indicate a deeper water transport entering the box below 1000 m depth. This deeper water must be mainly entering across line 70 because of the depth limitations for lines 67 and 77. This deeper water will be upwelling at the shore. As we observe on the potential temperature and salinity profiles, we have colder water shoreward. This colder water corresponds to the upwelling water that must be entering the box across line 70 below 1000 m.

There is another flow that we should take into account. As explained by



Commander (Mexican Navy) Juan Aguilar on his Master's Degree thesis presentation, offshore of the Monterey Bay there is a current flowing equatorward below 1300 m depth. This means that there is a large volume transport entering our box below 1000 m depth across line 67 (north side of the box), not computed in our work. This large deep transport across line 67 and the deep transport across line 70, responsible for the coastal upwelling, must be balancing the total 6.15 Sv leaving the box above 1000 m depth.



There is another factor that we should take into account if we want to clarify the excess in water transport leaving the box besides all the explained above. As explained earlier in this paper, there are some differences between the ADCP and the CTD velocity fields, even if we use the 250 m ADCP velocity as reference for the CTD computations. Above 150 m depth, there were differences between CTD and ADCP velocities due to wind transport and tidal currents.

We can compute the total transport for the differential velocity field obtained after subtracting CTD and ADCP velocities. The result of this residual transport computation is as follows:

Residual transport (obtained from CTD – ADCP velocity fields):

LINE 67:	-0.18 Sv
LINE 70:	+0.94 Sv
LINE 77:	+1.73 Sv

TOTAL:	+2.49 Sv (into the box)

As we can observe, the residual transport – not taken into account when using CTD velocity – is more important than expected, but is not large enough to balance the total transport leaving the box:

CTD Total transport:	-6.15 Sv
Residual transport:	+2.49 Sv
--	
Refined total transport:	-3.66 Sv

This results means that, as explained earlier, there is a large volume transport entering the box below 1000 db that is responsible for balancing the total transport obtained.

References

Colorado Center for Astrodynamics Research, “Global Near Real-Time Altimeter Data Viewer” [http://www-ccar.colorado.edu/~realtime/global-real-time_ssh/] August 2003.

Pond, S., and Pickard, G.L., *Introductory Dynamical Oceanography*, 2nd ed., pp 68-87, Butterworth/Heineman, 1983.

Pickard, G.L., and W.J. Emery, *Descriptive Physical Oceanography*, *An Introduction*, 5th ed., pp 95-120, Pergamon, 1990.

Aguilar, J., *Currents at the entrance to Monterey Bay*. Master’s Thesis. Naval Postgraduate Scool. Monterey. California. September 2003.

Broenkow, W., Moss Landing Marine Laboratories “The California Currents”, [http://www.mbnms.nos.noaa.gov/sitechar/phys21.html].. August 2003

Phillip P. Morgan, 1994. SEAWATER: A Library of MATLAB® Computational Routines for the Properties of Sea Water. CSIRO Marine Laboratories Report 222. 29 pp.

NOAA Coast Watch Product Search. Data Products NOAA-17 SST [http://coastwatch.noaa.gov/interface/interface.html] August 2003.

```
%       geospd.m
%       JUAN CONFORTO       AUG 2003
%       OC3570          CRUISE
%       PURPOSE: Compute and plot Geostrophic velocity, Salinity and Temperature
%       from Salinity, Pressure and Temperature obtained from CTD files.
%       This program uses CSIRO SeaWater subroutines.
%
%       FUNCTIONS USED:
%
%       sw_gpan = calculates geopotential anomaly
%       sw_gvel = calculates geostrophic velocity


%   Loops for loading each CTD matrix

clear all

for i=1:9
   loading = ['load r00' num2str(i) '.asc'];
   sizing = ['size(r00' num2str(i) ')'];
   eval(loading);
   M = eval(sizing);
   LINEAS(i) = M(1);   % Vector containing the number of lines of each CTD file

end

for i=10:31
   loading = ['load r0' num2str(i) '.asc'];
   sizing = ['size(r0' num2str(i) ')'];
   eval(loading);
   M = eval(sizing);
   LINEAS(i) = M(1);   % Vector containing the number of lines of each CTD file

end

%   Creating a vector of pressures in decibars (spaced 2 decibars like in CTD files)

for i=1:500
   P(i) = 2*i;
end

%   Creating vectors for Latitude and Longitude
%   and creating matrices for Salinity and Temperature

for i=1:31
   l = (LINEAS(i)+1)/2;
   if l>500;
      l=500;
   end

   if i==1;
   funcion1 = ['lat(i)= r00' num2str(i) '(1,2);'];
   funcion2 = ['lon(i)= r00' num2str(i) '(1,3);'];
   funcion3 = ['S(1:l,i)= r00' num2str(i) '(1:l,18);'];
   funcion4 = ['T(1:l,i)= r00' num2str(i) '(1:l,6);'];
   eval(funcion1);
```

```
   eval(funcion2);
   eval(funcion3);
   eval(funcion4);

   elseif i>1 & i<10;
   funcion1 = ['lat(i)= r00' num2str(i) '(1,2);'];
   funcion2 = ['lon(i)= r00' num2str(i) '(1,3);'];
   funcion3 = ['S(1:l,i)= r00' num2str(i) '(1:l,17);'];
   funcion4 = ['T(1:l,i)= r00' num2str(i) '(1:l,6);'];
   eval(funcion1);
   eval(funcion2);
   eval(funcion3);
   eval(funcion4);

   else i>10;
   funcion1 = ['lat(i)= r0' num2str(i) '(1,2);'];
   funcion2 = ['lon(i)= r0' num2str(i) '(1,3);'];
   funcion3 = ['S(1:l,i)= r0' num2str(i) '(1:l,17);'];
   funcion4 = ['T(1:l,i)= r0' num2str(i) '(1:l,6);'];
   eval(funcion1);
   eval(funcion2);
   eval(funcion3);
   eval(funcion4);
   end
end

P=P';   %  transposing Pressure vector


%   Here we use Seawater subroutines sw_gpan & sw_gvel

 ga = sw_gpan(S,T,P);    %   Geopotential anomaly

 vel = sw_gvel(ga,lat,lon);  %   Geostrophic velocity


%   Eliminating wrong values from vel Matrix (will produce vel=0
%   when there's no CTD value for that depth)

for i=1:30
   for j=1:500
      if S(j,i)==0;
         vel(j,i)=0;
      elseif S(j,i+1)==0;
         vel(j,i)=0;
      else
         vel(j,i)=vel(j,i);
      end
   end
end

%   changing the reference pressure level from surface
%   to the 1000 db level (=row #500).

velmil = [];
for i=1:30
```

```
   velmil(:,i)= vel(:,i) - vel(500,i);
end
for i=1:31
   l = (LINEAS(i)+1)/2;
   if l>500;
      l=500;
   end
   gamil(:,i)= ga(:,i) - ga(l,i);
end

%   Eliminating wrong values from vel Matrix (will produce vel=NaN
%   S=NaN & T=NaN when there's no CTD value for that depth)

for i=1:30
   for j=1:500
      if S(j,i)<29;
         S(j,i)= NaN;
         T(j,i)= NaN;
         velmil(j,i)=NaN;
         gamil(j,i)=NaN;
      elseif S(j,i+1)==0;
         velmil(j,i)=NaN;
      else
         velmil(j,i)=velmil(j,i);
      end
   end
end
for j=1:500
      if S(j,31)<29;
         S(j,31)= NaN;
         T(j,31)= NaN;
         gamil(j,31)=NaN;
      end
end

for i=1:30
   CTDstat(i)=i;   %  vector with the # of CTD station
end

 %        PLOTTING  RELATIVE GEOSTROPHIC VELOCITY

figure
contvelmil = [-.5:0.05:.4];
contourf(CTDstat,P,velmil,contvelmil);
axis('ij');
xlabel('(values between indicated and next CTD stations)')
ylabel('Pressure (Decibars)')
colorbar('vert')
shading flat;
title('                                    RELATIVE GEOSTROPHIC VELOCITY (m/s) ')
 %        PLOTTING GEOPOTENTIAL ANOMALY

CTDstat(31)=31;   %  vector with the # of CTD station
figure
contgamil = [-16:1:-5];
contourf(CTDstat,P,gamil,contgamil);
```

```
axis('ij');
xlabel('(values between indicated and next CTD stations)')
ylabel('Pressure (Decibars)')
colorbar('vert')
shading flat;
title('                         GEOPOTENTIAL ANOMALY (m^2 s^-2) ')


%       PLOTTING  SALINITY AND TEMPERATURE

figure
contS = [32:0.1:34.5];
contourf(CTDstat,P,S,contS);
axis('ij');
xlabel('CTD STATIONS')
ylabel('Pressure (Decibars)')
caxis([32 34.5]);
colorbar('vert')
shading flat;
title('                         SALINITY (PSU) ')

figure
contT = [2:0.5:18];
contourf(CTDstat,P,T,contT);
axis('ij');
xlabel('CTD STATIONS')
ylabel('Pressure (Decibars)')
caxis([1 18]);
colorbar('vert')
shading flat;
title('                         TEMPERATURE (ºC) ')


%       PLOTTING  T-S DIAGRAM

figure
for i=1:31
   plot(S(:,i),T(:,i),'.')
   hold on
end
xlabel('SALINITY (psu)')
ylabel('TEMPERATURE (ºC)')
hold off

%       PLOTTING  RELATIVE GEOSTROPHIC VELOCITY UP TO 450 db

STAT=CTDstat(1:30);
save STAT STAT;
P450geost=P(1:225);
save P450geost P450geost;
velmilB=velmil(1:225,:);
save geospd450 velmilB;

figure
contvelmil = [-.6:.001:.4];
contourf(STAT,P450geost,velmilB,contvelmil);
```

```
axis('ij');
xlabel('(values between indicated and next CTD stations)')
ylabel('Pressure (Decibars)')
colorbar('vert')
shading flat;
title('                              RELATIVE GEOSTROPHIC VELOCITY (m/s) ')
```

```
%       Ageospd.m
%       JUAN CONFORTO       AUG 2003
%       OC3570          CRUISE
%       PURPOSE: Compute and plot a more accurate Geostrophic velocity  field
%       using the ADCP velocity obtained at 250 Decibars as the reference level.
%
%       The initial velocity field is computed from Salinity, Pressure
%       and Temperature data (from CTD files) using the surface as the
%       level of no motion.
%
%       Then, we use the 250 Decibars ADCP velocity field as the reference level.
%
%       This program uses CSIRO SeaWater subroutines.
%
%       FUNCTIONS USED:
%
%       sw_gpan = calculates geopotential anomaly
%       sw_gvel = calculates geostrophic velocity


%   Loops for loading each CTD matrix

clear all

for i=1:9
   loading = ['load r00' num2str(i) '.asc'];
   sizing = ['size(r00' num2str(i) ')'];
   eval(loading);
   M = eval(sizing);
   LINEAS(i) = M(1);   % Vector containing the number of lines of each CTD file

end


for i=10:31
   loading = ['load r0' num2str(i) '.asc'];
   sizing = ['size(r0' num2str(i) ')'];
   eval(loading);
   M = eval(sizing);
   LINEAS(i) = M(1);   % Vector containing the number of lines of each CTD file

end

%   Creating a vector of pressures in decibars (spaced 2 decibars like in CTD files)

for i=1:500
   P(i) = 2*i;
end

%   Creating vectors for Latitude and Longitude
%   and creating matrices for Salinity and Temperature

for i=1:31
   l = (LINEAS(i)+1)/2;
   if l>500;
      l=500;
```

```
   end

   if i==1;
   funcion1 = ['lat(i)= r00' num2str(i) '(1,2);'];
   funcion2 = ['lon(i)= r00' num2str(i) '(1,3);'];
   funcion3 = ['S(1:l,i)= r00' num2str(i) '(1:l,18);'];
   funcion4 = ['T(1:l,i)= r00' num2str(i) '(1:l,6);'];
   eval(funcion1);
   eval(funcion2);
   eval(funcion3);
   eval(funcion4);

   elseif i>1 & i<10;
   funcion1 = ['lat(i)= r00' num2str(i) '(1,2);'];
   funcion2 = ['lon(i)= r00' num2str(i) '(1,3);'];
   funcion3 = ['S(1:l,i)= r00' num2str(i) '(1:l,17);'];
   funcion4 = ['T(1:l,i)= r00' num2str(i) '(1:l,6);'];
   eval(funcion1);
   eval(funcion2);
   eval(funcion3);
   eval(funcion4);

   else i>10;
   funcion1 = ['lat(i)= r0' num2str(i) '(1,2);'];
   funcion2 = ['lon(i)= r0' num2str(i) '(1,3);'];
   funcion3 = ['S(1:l,i)= r0' num2str(i) '(1:l,17);'];
   funcion4 = ['T(1:l,i)= r0' num2str(i) '(1:l,6);'];
   eval(funcion1);
   eval(funcion2);
   eval(funcion3);
   eval(funcion4);
   end
end

P=P';   %  transposing Pressure vector



%   Here we use Seawater subroutines sw_gpan & sw_gvel

 ga = sw_gpan(S,T,P);    %   Geopotential anomaly

 vel = sw_gvel(ga,lat,lon);  %   Geostrophic velocity


%   Eliminating wrong values from vel Matrix (will produce vel=0
%   when there's no CTD value for that depth)

for i=1:30
   for j=1:500
      if S(j,i)==0;
         vel(j,i)=0;
      elseif S(j,i+1)==0;
         vel(j,i)=0;
      else
         vel(j,i)=vel(j,i);
```

```
    end
  end
end


%   LOADING THE FILE OF THE REFERENCE LEVEL ACTUAL VELOCITIES

load ADCP250;
ADCP250=ADCP250./100;

%   changing the level no motion from surface
%   to the 250 db level (=row #125).
%   ...and

vel250 = [];
for i=1:30
  vel250(:,i)= vel(:,i) - vel(125,i);
  ACCvel(:,i)=vel250(:,i)+ADCP250(1,i);
end

for i=1:31
  l = (LINEAS(i)+1)/2;
  if l>500;
    l=500;
  end
  gamil(:,i)= ga(:,i) - ga(l,i);
end


%   Eliminating wrong values from vel Matrix (will produce vel=NaN
%   S=NaN & T=NaN when there's no CTD value for that depth)

for i=1:30
  for j=1:500
    if S(j,i)<29;
      S(j,i)= NaN;
      T(j,i)= NaN;
      ACCvel(j,i)=NaN;
      gamil(j,i)=NaN;
    elseif S(j,i+1)==0;
      ACCvel(j,i)=NaN;
    else
      ACCvel(j,i)=ACCvel(j,i);
    end
  end
end
for j=1:500
    if S(j,31)<29;
      S(j,31)= NaN;
      T(j,31)= NaN;
      gamil(j,31)=NaN;
    end
end

for i=1:30
  CTDstat(i)=i;   %  vector with the # of CTD station
```

```
end

 %       PLOTTING  RELATIVE GEOSTROPHIC VELOCITY

figure
contACCvel = [-.6:0.01:.4];
contourf(CTDstat,P,ACCvel,contACCvel);
axis('ij');
xlabel('(values between indicated and next CTD stations)')
ylabel('Pressure (Decibars)')
colorbar('vert')
shading flat;
title('                RELATIVE GEOSTROPHIC VELOCITY (m/s) COMPUTED USING THE ADCP
VELOCITY AT 250 DECIBARS')

%       PLOTTING  RELATIVE GEOSTROPHIC VELOCITY UP TO 450 db

STAT=CTDstat(1:30);
save STAT STAT;
P450geost=P(1:225);
save P450geost P450geost;
ACCvelB=ACCvel(1:225,:);
save ACCgeospd450 ACCvelB;

figure
contACCvel = [-.6:.01:.4];
contourf(STAT,P450geost,ACCvelB,contACCvel);
axis('ij');
xlabel('(values between indicated and next CTD stations)')
ylabel('Pressure (Decibars)')
colorbar('vert')
shading flat;
title('                RELATIVE GEOSTROPHIC VELOCITY (m/s) COMPUTED USING THE ADCP
VELOCITY AT 250 DECIBARS')
```

```
%       geospdST.m
%       JUAN CONFORTO       AUG 2003
%       OC3570          CRUISE
%       PURPOSE: Compute and plot Geostrophic velocity, Salinity and Temperature
%       from Salinity, Pressure and Temperature obtained from CTD files.
%       This program uses CSIRO SeaWater subroutines.
%
%       FUNCTIONS USED:
%
%       sw_gpan = calculates geopotential anomaly
%       sw_gvel = calculates geostrophic velocity


%   Loops for loading each CTD matrix

clear all

for i=1:9
   loading = ['load r00' num2str(i) '.asc'];
   sizing = ['size(r00' num2str(i) ')'];
   eval(loading);
   M = eval(sizing);
   LINEAS(i) = M(1);   % Vector containing the number of lines of each CTD file

end


for i=10:31
   loading = ['load r0' num2str(i) '.asc'];
   sizing = ['size(r0' num2str(i) ')'];
   eval(loading);
   M = eval(sizing);
   LINEAS(i) = M(1);   % Vector containing the number of lines of each CTD file

end

%   Creating a vector of pressures in decibars (spaced 2 decibars like in CTD files)

for i=1:500
   P(i) = 2*i;
end

%   Creating vectors for Latitude and Longitude
%   and creating matrices for Salinity and Temperature

for i=1:31
   l = (LINEAS(i)+1)/2;
   if l>500;
      l=500;
   end

   if i==1;
   funcion1 = ['lat(i)= r00' num2str(i) '(1,2);'];
   funcion2 = ['lon(i)= r00' num2str(i) '(1,3);'];
   funcion3 = ['S(1:l,i)= r00' num2str(i) '(1:l,18);'];
   funcion4 = ['T(1:l,i)= r00' num2str(i) '(1:l,6);'];
```

```
    eval(funcion1);
    eval(funcion2);
    eval(funcion3);
    eval(funcion4);

    elseif i>1 & i<10;
    funcion1 = ['lat(i)= r00' num2str(i) '(1,2);'];
    funcion2 = ['lon(i)= r00' num2str(i) '(1,3);'];
    funcion3 = ['S(1:l,i)= r00' num2str(i) '(1:l,17);'];
    funcion4 = ['T(1:l,i)= r00' num2str(i) '(1:l,6);'];
    eval(funcion1);
    eval(funcion2);
    eval(funcion3);
    eval(funcion4);

    else i>10;
    funcion1 = ['lat(i)= r0' num2str(i) '(1,2);'];
    funcion2 = ['lon(i)= r0' num2str(i) '(1,3);'];
    funcion3 = ['S(1:l,i)= r0' num2str(i) '(1:l,17);'];
    funcion4 = ['T(1:l,i)= r0' num2str(i) '(1:l,6);'];
    eval(funcion1);
    eval(funcion2);
    eval(funcion3);
    eval(funcion4);
    end
end

P=P';   %  transposing Pressure vector



%   Here we use Seawater subroutines sw_gpan & sw_gvel

 ga = sw_gpan(S,T,P);    %   Geopotential anomaly
 distkm = 1000*sw_dist(lat,lon,'km');
 vel = sw_gvel(ga,lat,lon);  %   Geostrophic velocity

%    producing a vector of distances

dist(1,1)=distkm(1,1);
for d=2:30
    dist(1,d)=dist(1,d-1)+distkm(1,d-1);
end

save dist dist;

%   Eliminating wrong values from vel Matrix (will produce vel=0
%   when there's no CTD value for that depth)

for i=1:30
    for j=1:500
        if S(j,i)==0;
            vel(j,i)=0;
        elseif S(j,i+1)==0;
            vel(j,i)=0;
        else
```

```
            vel(j,i)=vel(j,i);
        end
    end
end

%   changing the reference pressure level from surface
%   to the 1000 db level (=row #500).

velmil = [];
for i=1:30
    velmil(:,i)= vel(:,i) - vel(500,i);
end
for i=1:31
    l = (LINEAS(i)+1)/2;
    if l>500;
        l=500;
    end
    gamil(:,i)= ga(:,i); %- ga(l,i);
end

%   Eliminating wrong values from vel Matrix (will produce vel=NaN
%   S=NaN & T=NaN when there's no CTD value for that depth)

for i=1:30
    for j=1:500
        if S(j,i)<29;
            S(j,i)= NaN;
            T(j,i)= NaN;
            velmil(j,i)=NaN;
            gamil(j,i)=NaN;
        elseif S(j,i+1)==0;
            velmil(j,i)=NaN;
        else
            velmil(j,i)=velmil(j,i);
        end
    end
end
for j=1:500
        if S(j,31)<29;
            S(j,31)= NaN;
            T(j,31)= NaN;
            gamil(j,31)=NaN;
        end
end

for i=1:30
    CTDstat(i)=i;   %  vector with the # of CTD station
end


%   computing potential temperature
PT = sw_ptmp(S,T,P,0);
DT = PT - T;

%   plotting Temperature and Salinity profiles for each station
%   (grouped in legs)
```

```
figure
for i=1:10
   ct=2*i-1;
   cs=2*i;
   subplot(5,4,ct)
plot(T(:,i),P);
axis('ij')
title(' Pot Temp')
subplot(5,4,cs)
plot(S(:,i),P);
axis('ij');
title(' Salinity')
end

figure
for j=11:22
   i=j-10;
   ct=2*i-1;
   cs=2*i;
   subplot(6,4,ct)
plot(PT(:,j),P);
axis('ij')
title(' Pot Temp')
subplot(6,4,cs)
plot(S(:,j),P);
axis('ij');
title(' Salinity')
end

figure
for j=23:31
   i=j-22;
   ct=2*i-1;
   cs=2*i;
   subplot(3,6,ct)
plot(PT(:,j),P);
axis('ij')
title(' Pot Temp')
subplot(3,6,cs)
plot(S(:,j),P);
axis('ij');
title(' Salinity')
end
```

```
%       Ageotransport.m
%       JUAN CONFORTO       AUG 2003
%       OC3570          CRUISE
%       PURPOSE: Compute and plot a more accurate Transport
%       using the ADCP velocity obtained at 250 Decibars as the reference level.
%
%       The initial velocity field is computed from Salinity, Pressure
%       and Temperature data (from CTD files) using the surface as the
%       level of no motion.
%
%       Then, we use the 250 Decibars ADCP velocity field as the refernce level.
%
%       This program uses CSIRO SeaWater subroutines.
%
%       FUNCTIONS USED:
%
%       sw_gpan = calculates geopotential anomaly
%       sw_gvel = calculates geostrophic velocity


%   Loops for loading each CTD matrix

clear all

for i=1:9
   loading = ['load r00' num2str(i) '.asc'];
   sizing = ['size(r00' num2str(i) ')'];
   eval(loading);
   M = eval(sizing);
   LINEAS(i) = M(1);   % Vector containing the number of lines of each CTD file

end


for i=10:31
   loading = ['load r0' num2str(i) '.asc'];
   sizing = ['size(r0' num2str(i) ')'];
   eval(loading);
   M = eval(sizing);
   LINEAS(i) = M(1);   % Vector containing the number of lines of each CTD file

end

%   Creating a vector of pressures in decibars (spaced 2 decibars like in CTD files)

for i=1:500
   P(i) = 2*i;
end

%   Creating vectors for Latitude and Longitude
%   and creating matrices for Salinity and Temperature

for i=1:31
   l = (LINEAS(i)+1)/2;
   if l>500;
      l=500;
```

```
   end

   if i==1;
   funcion1 = ['lat(i)= r00' num2str(i) '(1,2);'];
   funcion2 = ['lon(i)= r00' num2str(i) '(1,3);'];
   funcion3 = ['S(1:l,i)= r00' num2str(i) '(1:l,18);'];
   funcion4 = ['T(1:l,i)= r00' num2str(i) '(1:l,6);'];
   eval(funcion1);
   eval(funcion2);
   eval(funcion3);
   eval(funcion4);

   elseif i>1 & i<10;
   funcion1 = ['lat(i)= r00' num2str(i) '(1,2);'];
   funcion2 = ['lon(i)= r00' num2str(i) '(1,3);'];
   funcion3 = ['S(1:l,i)= r00' num2str(i) '(1:l,17);'];
   funcion4 = ['T(1:l,i)= r00' num2str(i) '(1:l,6);'];
   eval(funcion1);
   eval(funcion2);
   eval(funcion3);
   eval(funcion4);

   else i>10;
   funcion1 = ['lat(i)= r0' num2str(i) '(1,2);'];
   funcion2 = ['lon(i)= r0' num2str(i) '(1,3);'];
   funcion3 = ['S(1:l,i)= r0' num2str(i) '(1:l,17);'];
   funcion4 = ['T(1:l,i)= r0' num2str(i) '(1:l,6);'];
   eval(funcion1);
   eval(funcion2);
   eval(funcion3);
   eval(funcion4);
   end
end

P=P';   %  transposing Pressure vector



%   Here we use Seawater subroutines sw_gpan & sw_gvel

 ga = sw_gpan(S,T,P);    %   Geopotential anomaly
 dist_m = 1000*sw_dist(lat,lon,'km'); %    Distance between stations in Kilometers
 vel = sw_gvel(ga,lat,lon);  %   Geostrophic velocity
 profundidad = sw_dpth(P,lat(1));       %   converts pressure to depth in meters

deltadepth = profundidad(2) - profundidad(1);


%   Eliminating wrong values from vel Matrix (will produce vel=0
%   when there's no CTD value for that depth)

for i=1:30
   for j=1:500
      if S(j,i)==0;
         vel(j,i)=0;
      elseif S(j,i+1)==0;
```

```
        vel(j,i)=0;
      else
        vel(j,i)=vel(j,i);
      end
  end
end


%   LOADING THE FILE OF THE REFERENCE LEVEL ACTUAL VELOCITIES

load ADCP250;
ADCP250=ADCP250./100;

%   changing the level no motion from surface
%   to the 250 db level (=row #125).
%   ...and

vel250 = [];
for i=1:30
  vel250(:,i)= vel(:,i) - vel(125,i);
  ACCvel(:,i)=vel250(:,i)+ADCP250(1,i);
end

for i=1:31
  l = (LINEAS(i)+1)/2;
  if l>125;
    l=125;
  end
  gamil(:,i)= ga(:,i) - ga(l,i);
end


%   Eliminating wrong values from vel Matrix (will produce vel=NaN
%   S=NaN & T=NaN when there's no CTD value for that depth)

for i=1:30
  for j=1:500
    if S(j,i)<29;
      S(j,i)= NaN;
      T(j,i)= NaN;
      ACCvel(j,i)=NaN;
      gamil(j,i)=NaN;
    elseif S(j,i+1)==0;
      ACCvel(j,i)=NaN;
    else
      ACCvel(j,i)=ACCvel(j,i);
    end
  end
end
for j=1:500
    if S(j,31)<29;
      S(j,31)= NaN;
      T(j,31)= NaN;
      gamil(j,31)=NaN;
    end
end
```

```
for i=1:30
   CTDstat(i)=i;   %  vector with the # of CTD station
end

%   Computing the geostrophic velocity volume transport

for i=1:30
   voltrans(:,i) = (ACCvel(:,i)*deltadepth*dist_m(i))./(1.0e+6);
end



 %        PLOTTING  VOLUME TRANSPORT

figure
contourf(CTDstat,profundidad,voltrans);
axis('ij');
xlabel('(values between indicated and next CTD stations)')
ylabel('depth (m)')
colorbar('vert')
shading flat;
title('              VOLUME TRANSPORT OBTAINED FROM RELATIVE GEOSTROPHIC
VELOCITY (m^3/s) ')


%   plotting volume transport profiles for each pair of stations
%   (grouped in legs)

figure
for i=1:9
   subplot(3,3,i)
plot(voltrans(:,i),profundidad);
axis('ij');
end

figure
N=0
for i=10:21
   N=N+1
   subplot(4,3,N)
plot(voltrans(:,i),profundidad);
axis('ij');
end

figure
N=0
for i=22:30
   N=N+1
   subplot(3,3,N)
plot(voltrans(:,i),profundidad);
axis('ij');
end

%   computing total transport
voltrans2 = voltrans;
```

```
for i=1:30
   for j=1:500
      if voltrans(j,i) > -50e+6 | voltrans(j,i) < 50e+6;
         voltrans2(j,i)= voltrans(j,i);
      else
         voltrans2(j,i)= 0;
      end
   end
end

leg1 = voltrans2(:,1:9);
leg2 = voltrans2(:,10:21);
leg3 = voltrans2(:,22:30);

tot_transX = sum(leg1);
tot_trans1 = sum(tot_transX)

tot_transX = sum(leg2);
tot_trans2 = sum(tot_transX)

tot_transX = sum(leg3);
tot_trans3 = sum(tot_transX)

%   plotting transport profile between each pair of stations

for i=1:30
   disp('columna')
   i
   sum(voltrans2(:,i))
end
```

```
%       adcp.m
%       AUTHOR: Curt Collins august 16 2001
%       MODIFIED BY: JUAN CONFORTO      AUG 2003
%       OC3570          CRUISE
%       PURPOSE: Load and plot ADCP data
%
%
%        data file box.con is produced by copying rot67.con + rot70.con + rot77.con

clear all
close all

load ROT67.CON;
x1=ROT67(:,1);
n1=length(x1);
clear ROT67; clear x1;
load ROT70.con;
x2=ROT70(:,1);
n2=length(x2);
clear ROT70; clear x2;
load ROT77.con;
x3=ROT77(:,1);
n3=length(x3);
clear ROT77; clear x3;

load box.con
x=box(:,1);z=box(:,2);u=box(:,3);v=box(:,4);
n=length(x);
for i=1:n1
dist(i)=(238.1302-x(i))*51.5*1.852;
uo(i)=u(i); vo(i)=v(i);
end
dist1=dist(n1)+((5.15+6.9)*1.852)/2;

for j=n1+1:n1+n2
dist(j)=dist1 + (36.1209-x(j))*69*1.852;
uo(j)=v(j); vo(j)=-u(j);
end;
dist2=dist(n1+n2)+((5.15+6.9)*1.852)/2;

for k=n1+n2+1:n1+n2+n3
dist(k)=dist2 + (x(k)-237.7576)*51.5*1.852;
uo(k)=-u(k); vo(k)=-v(k);
end;

fid=fopen('box.dat','w+');

for k = 1:n
format1='%9.1f %9.1f %9.1f %9.1f \n';
fprintf(fid,format1,dist(k),z(k),uo(k),vo(k));
end
fclose(fid);


save box -ascii
```

```
K=size(box);
DL = K(1)-1;

%   CREATING A VECTOR FOR DISTANCES (L) AND
%   MATRICES FOR U AND V

r=1;
c=1;
for t=1:DL
   L(c)=box(t,1);
   U(r,c)=box(t,3);
   V(r,c)=box(t,4);
   if box(t,1) ~= box(t+1,1);
      c=c+1;
      r=1;
   else
      r=r+1;
   end
end

J=size(U);
JR=J(1);
JC=J(2);

for t=1:JR
   D(t)=10*(t+1);
end

%D=D';

for t=1:JC
   ST(t)=t;
end

%   Eliminating wrong values from U & V Matrices (will produce U=NaN
%   & V=NaN when there's no ADCP value for that depth)

for i=1:JC
   for j=1:JR
      if U(j,i)==0;
         U(j,i)= NaN;
         V(j,i)= NaN;
      end
   end
end


%   SAVING DATA

save U U;
save V V;
save D D;
save ST ST;


 %       PLOTTING U ADCP VELOCITY
```

```
figure
contU = [-60:1:40];
contourf(ST,D,U,contU);
axis('ij');
xlabel('ADCP STATIONS')
ylabel('depth (m)')
colorbar('vert')
shading flat;
title('                              U VELOCITY (cm/s) ')

  %      PLOTTING V ADCP VELOCITY

figure
contV = [-60:1:40];
contourf(ST,D,V,contV);
axis('ij');
xlabel('ADCP STATIONS')
ylabel('depth (m)')
colorbar('vert')
shading flat;
title('                              V VELOCITY (cm/s) ')
```

```
%	veldif.m
%	JUAN CONFORTO	AUG 2003
%	OC3570		CRUISE
%	PURPOSE: compute the difference between CTD-obtained geostrophic
%	velocity and ADCP-measured velocity
%

clear all
close all

%   loading CTD-computed data
load P450geost.mat;
load STAT.mat;
load geospd450.mat;
load dist.mat;

%   loading ADCP-computed data
load D.mat;
load ST.mat;
load U.mat;
load V.mat;


%velmilB = velmilB';
velmilB = velmilB.*100;
D=D';
ST=ST';



dmax=max(dist);

sizeCTD=size(velmilB);
gspdstations=sizeCTD(1,2);

sizeADCP=size(U);
Ustations=sizeADCP(1,2);

distADCP=linspace(0,dmax,Ustations);

dist=dist./1000;
distADCP=distADCP./1000;
distADCP=distADCP';

U=U';


%   CREATING NEW MATRICES FOR THE ADCP U VELOCITY
%   AND FOR THE CTD GEOSTROPHIC VELOCITY WITH THE SAME
%   SIZE

ADCP = griddata(D,distADCP,U,D,dist);
ADCP=ADCP';

velmilB=velmilB';
CTD = griddata(P450geost,dist,velmilB,D,dist);
```

```
CTD=CTD';

ADCP250 = ADCP(24,:);

save ADCP250 ADCP250;

%   COMPUTING THE DIFFERENCE BETWEEN CTD GEOSTROPHIC VELOCITY
%   AND ADCP VELOCITY

DIF = CTD-ADCP;


 %       PLOTTING CTD geostrophic VELOCITY

figure
contDIF = [-60:1:40];
contourf(dist,D,CTD,contDIF);
axis('ij');
xlabel('distance (Km)')
ylabel('depth (m)')
colorbar('vert')
shading flat;
title('                              CTD geostrophic velocity (cm/s) ')

 %       PLOTTING U ADCP VELOCITY

figure
contDIF = [-60:1:40];
contourf(dist,D,ADCP,contDIF);
axis('ij');
xlabel('distance(Km)')
ylabel('depth (m)')
colorbar('vert')
shading flat;
title('                              ADCP measured U velocity (cm/s) ')

 %       PLOTTING U ADCP VELOCITY

figure
contDIF = [-60:1:40];
contourf(dist,D,DIF,contDIF);
axis('ij');
xlabel('distance (Km)')
ylabel('depth (m)')
colorbar('vert')
shading flat;
title('                              VELOCITY DIFFERENCES (cm/s) ')
```

```
%       Aveldif.m
%       JUAN CONFORTO      AUG 2003
%       OC3570        CRUISE
%       PURPOSE: compute the difference between CTD-obtained geostrophic
%       velocity (obtained using 250 m ADCP velocity as a reference)
%       and ADCP-measured velocity
%

clear all
close all

%   loading CTD-computed data
load P450geost.mat;
load STAT.mat;
load ACCgeospd450.mat;
load dist.mat;

%   loading ADCP-computed data
load D.mat;
load ST.mat;
load U.mat;
load V.mat;

ACCvelB = ACCvelB.*100;
D=D';
ST=ST';



dmax=max(dist);

sizeCTD=size(ACCvelB);
gspdstations=sizeCTD(1,2);

sizeADCP=size(U);
Ustations=sizeADCP(1,2);

distADCP=linspace(0,dmax,Ustations);

dist=dist./1000;
distADCP=distADCP./1000;
distADCP=distADCP';

U=U';


%   CREATING NEW MATRICES FOR THE ADCP U VELOCITY
%   AND FOR THE CTD GEOSTROPHIC VELOCITY WITH THE SAME
%   SIZE

ADCP = griddata(D,distADCP,U,D,dist);
ADCP=ADCP';

ACCvelB=ACCvelB';
CTD = GRIDDATA(P450geost,dist,ACCvelB,D,dist);
CTD=CTD';
```

```
ADCP250 = ADCP(24,:);

save ADCP250 ADCP250;

%   COMPUTING THE DIFFERENCE BETWEEN CTD GEOSTROPHIC VELOCITY
%   AND ADCP VELOCITY

DIF = CTD-ADCP;


 %       PLOTTING CTD geostrophic VELOCITY

figure
contDIF = [-60:1:40];
contourf(dist,D,CTD,contDIF);
axis('ij');
xlabel('distance (Km)')
ylabel('depth (m)')
colorbar('vert')
shading flat;
title('                              CTD geostrophic velocity (cm/s) ')

 %       PLOTTING U ADCP VELOCITY

figure
contDIF = [-60:1:40];
contourf(dist,D,ADCP,contDIF);
axis('ij');
xlabel('distance(Km)')
ylabel('depth (m)')
colorbar('vert')
shading flat;
title('                              ADCP measured U velocity (cm/s) ')

 %       PLOTTING U ADCP VELOCITY

figure
contDIF = [-60:1:40];
contourf(dist,D,DIF,contDIF);
axis('ij');
xlabel('distance (Km)')
ylabel('depth (m)')
colorbar('vert')
shading flat;
title('VELOCITY DIFFERENCES (cm/s) (computed more accurately using the 250 db ADCP
velocity as reference level for CTD geostrophic velocity) ')
```

```
%       AveldifTRANSPORT.m
%       JUAN CONFORTO       AUG 2003
%       OC3570          CRUISE
%       PURPOSE: This program computes the residual transport due to wind (Ekman)
%       and Tides, not taken into account when computing transport using
%       the CTD geostrophic velocity field.
%
%       This program first computes the difference between CTD-obtained geostrophic
%       velocity and ADCP-measured velocity. This difference velocity field is mostly
%       due to Ekman transport (wind) and Tide transport. Using this differences field
%       the residual transport is computed for lines 67, 70 and 77, and for the whole
%       67+70+77 line.
%

clear all
close all

%   loading CTD-computed data
load P450geost.mat;
load STAT.mat;
load ACCgeospd450.mat;
load dist.mat;

%   loading ADCP-computed data
load D.mat;
load ST.mat;
load U.mat;
load V.mat;

ACCvelB = ACCvelB.*100;
D=D';
ST=ST';



dmax=max(dist);

sizeCTD=size(ACCvelB);
gspdstations=sizeCTD(1,2);

sizeADCP=size(U);
Ustations=sizeADCP(1,2);

distADCP=linspace(0,dmax,Ustations);

dist=dist./1000;
distADCP=distADCP./1000;
distADCP=distADCP';

U=U';


%   CREATING NEW MATRICES FOR THE ADCP U VELOCITY
%   AND FOR THE CTD GEOSTROPHIC VELOCITY WITH THE SAME
%   SIZE
```

```
ADCP = griddata(D,distADCP,U,D,dist);
ADCP=ADCP';

ACCvelB=ACCvelB';
CTD = GRIDDATA(P450geost,dist,ACCvelB,D,dist);
CTD=CTD';

ADCP250 = ADCP(24,:);

save ADCP250 ADCP250;

%   COMPUTING THE DIFFERENCE BETWEEN CTD GEOSTROPHIC VELOCITY
%   AND ADCP VELOCITY

DIF = CTD-ADCP;

for i=1:30
   CTDstat(i)=i;   %  vector with the # of CTD station
end

%   Computing the residual (differencial) velocity volume transport

for i=1:30
   voltrans(:,i) = (DIF(:,i)*10*dist(i))./(1.0e+6);
end



%   plotting volume transport profiles for each pair of stations
%   (grouped in legs)

figure
for i=1:9
   subplot(3,3,i)
plot(voltrans(:,i),D);
axis('ij');
end

figure
N=0
for i=10:21
   N=N+1
   subplot(4,3,N)
plot(voltrans(:,i),D);
axis('ij');
end

figure
N=0
for i=22:30
   N=N+1
   subplot(3,3,N)
plot(voltrans(:,i),D);
axis('ij');
end
```

```
% computing total transport
voltrans2 = voltrans;
for i=1:30
   for j=1:45
      if voltrans(j,i) > -50e+6 | voltrans(j,i) < 50e+6;
         voltrans2(j,i)= voltrans(j,i);
      else
         voltrans2(j,i)= 0;
      end
   end
end

leg1 = voltrans2(:,1:9);
leg2 = voltrans2(:,10:21);
leg3 = voltrans2(:,22:30);

tot_transX = sum(leg1);
tot_trans1 = sum(tot_transX)

tot_transX = sum(leg2);
tot_trans2 = sum(tot_transX)

tot_transX = sum(leg3);
tot_trans3 = sum(tot_transX)

tot_transX = sum(voltrans2);
tot_transTOTAL = sum(tot_transX)


%for i=1:30
 %   disp('columna')
 %   i
 %   sum(voltrans2(:,i))
 %end
```